SOEN2070 C++程序设计

09 流

刘安 anliu@suda.edu.cn 2023-2024-2

流

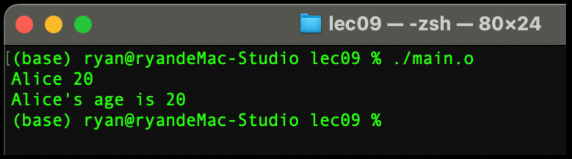
- 程序的基本任务：接受输入、产生输出
- 流指明数据的来源和目的地
- 标准库支持
  - 标准流：控制台
    - 图形用户界面GUI应用程序通常没有控制台
  - 文件流：文件
  - 字符串流：字符串

标准流

- cin - 输入流，从控制台读取用户输入
- cout - 缓冲的输出流， 向控制台写入数据
- cerr - 无缓冲的输出流，向错误控制台（通常就是控制台）写入数据
- clog - 有缓冲的cerr

标准输入流cin

```
string name;
int age;
cin >> name >> age;
cout << name << "'s age is " << age << endl;
```

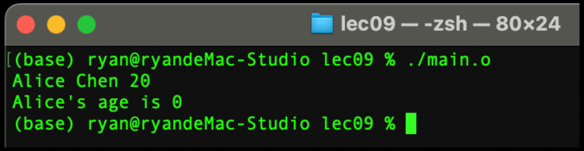


```
(base) ryan@ryandeMac-Studio lec09 % ./main.o
Alice 20
Alice's age is 20
(base) ryan@ryandeMac-Studio lec09 %
```

标准输入流cin

```
string name;
int age;
cin >> name >> age;
cout << name << "'s age is " << age << endl;
```

当遇到任意空白字符或者针对当前类型来说不合法的字符时，运算符>>停止读入

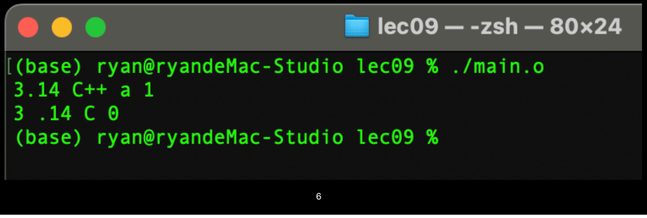


```
(base) ryan@ryandeMac-Studio lec09 % ./main.o
Alice Chen 20
Alice's age is 0
(base) ryan@ryandeMac-Studio lec09 %
```

标准输入流cin

```
int a; string b; char c; bool d;
cin >> a >> b >> c >> d; //user type: 3.14 C++ a 1
cout << a << ' ' << b << ' ' << c << ' ' << d << endl;
```

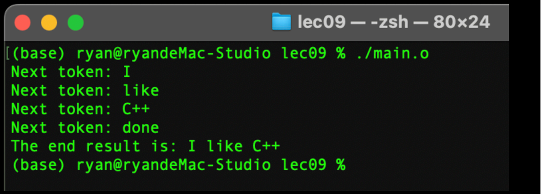
当遇到任意空白字符或者针对当前类型来说不合法的字符时，运算符>>停止读入



```
(base) ryan@ryandeMac-Studio lec09 % ./main.o
3.14 C++ a 1
3 .14 C 0
(base) ryan@ryandeMac-Studio lec09 %
```

字符串流

- istringstream从string中读出数据，ostringstream将数据写入string
- #include <sstream>



```
(base) ryan@ryandeMac-Studio lec09 % ./main.o
Next token: I
Next token: like
Next token: C++
Next token: done
The end result is: I like C++
(base) ryan@ryandeMac-Studio lec09 %
```

```
ostringstream oss;
while (cin) {
    string next_token;
    cout << "Next token: ";
    cin >> next_token;
    if (next_token == "done") break;
    oss << next_token << ' ';
}
cout << "The end result is: " << oss.str() << endl;
```

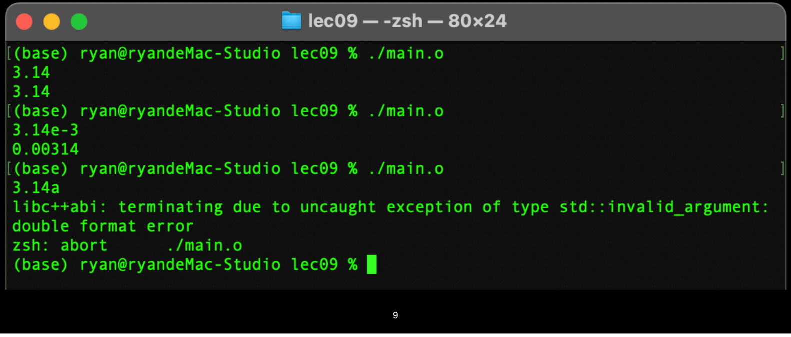
字符串流

- istringstream从string中读出数据，ostringstream将数据写入string
- #include <sstream>

```
double str_to_double(string s)
//如果可能，将字符串转换为浮点数
{
    istringstream iss { s }; //定义一个流从s中读取数据
    double d;
    iss >> d;
    if (!iss) throw invalid_argument("double format error");
    return d;
}
```

字符串流

```
string s;
cin >> s;
cout << str_to_double(s) << endl;
```



```
[(base) ryan@ryandeMac-Studio lec09 % ./main.o                                  ]
3.14
3.14
[(base) ryan@ryandeMac-Studio lec09 % ./main.o                                  ]
3.14e-3
0.00314
[(base) ryan@ryandeMac-Studio lec09 % ./main.o                                  ]
3.14a
libc++abi: terminating due to uncaught exception of type std::invalid_argument:
double format error
zsh: abort      ./main.o
(base) ryan@ryandeMac-Studio lec09 % █
```

面向行的输入

- 当遇到任意空白字符或针对当前类型来说不合法的字符时，运算符>>停止读入
- 可以使用getline函数一次读取整行内容（包括空白字符）
- 然后再通过输入字符串流istringstream从string中读取数据

```
string s;
std::getline(cin, s); //user type: Alice Chen 20
cout << s << endl;
istringstream iss { s };
string first_name, last_name; int age;
iss >> first_name >> last_name >> age;
cout << first_name << "'s age is: " << age << endl;
```

文件流

- ifstream从文件中读出数据，ofstream将数据写入文件
- #include <fstream>

```
cout << "Please enter input file name: ";
string in_name;
cin >> in_name;
//ist是以in_name命名的文件对应的输入流
ifstream ist { in_name };
if (!ist) throw runtime_error("cannot open input file");
//从文件流读取数据（见下页）
```

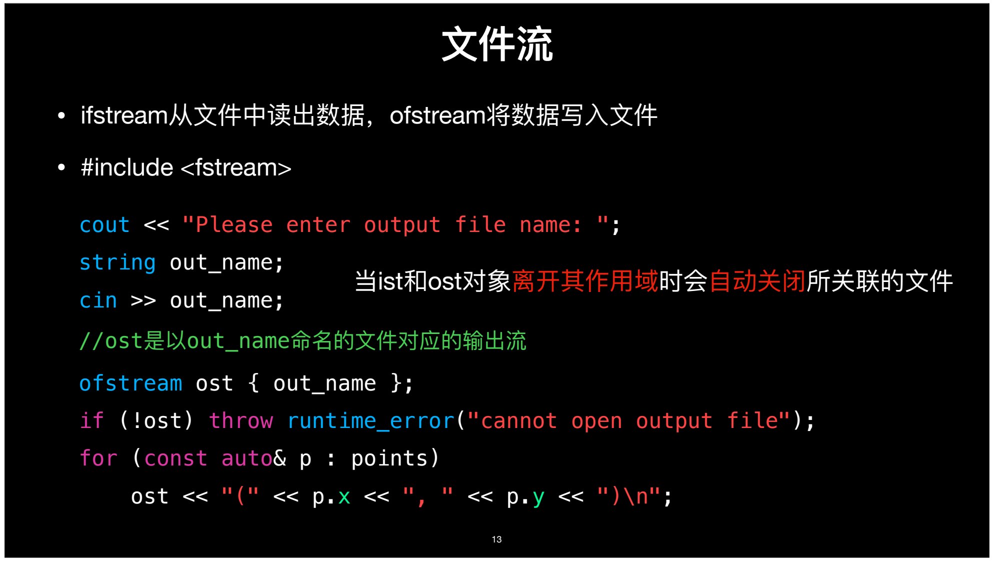
文件流

- ifstream从文件中读出数据，ofstream将数据写入文件
- #include <fstream>

```
vector<Point> points;
for (Point p; ist >> p; )
    points.push_back(p);
// 将points写入文件，见下页
```

```
struct Point {
    double x, y;
};

istream&
operator>>(istream& is, Point& p)
{
    is >> p.x >> p.y;
    return is;
}
```

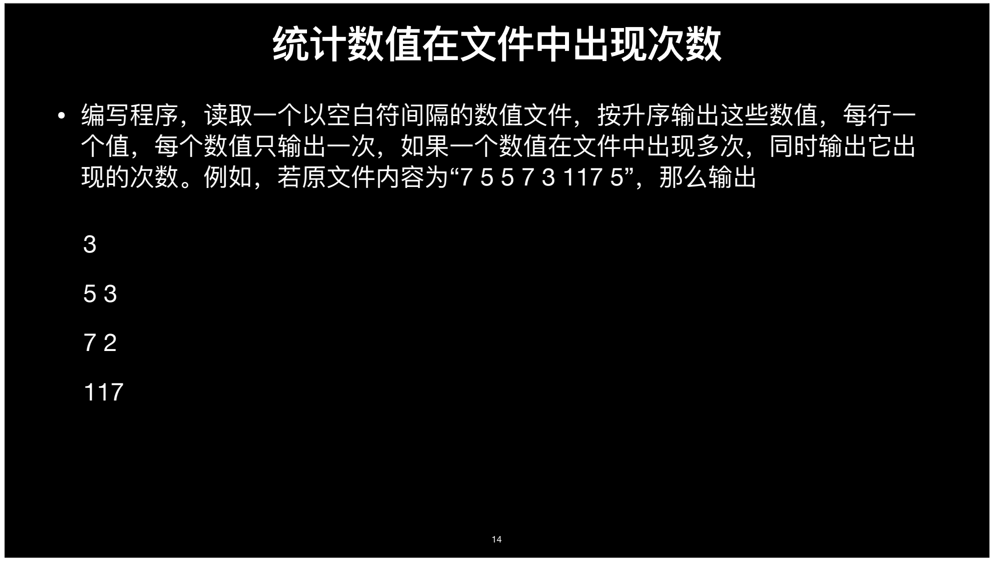


文件流

- ifstream从文件中读出数据，ofstream将数据写入文件
- #include <fstream>

```
cout << "Please enter output file name: ";
string out_name;
cin >> out_name;
//ost是以out_name命名的文件对应的输出流
ofstream ost { out_name };
if (!ost) throw runtime_error("cannot open output file");
for (const auto& p : points)
    ost << "(" << p.x << ", " << p.y << ")\n";
```

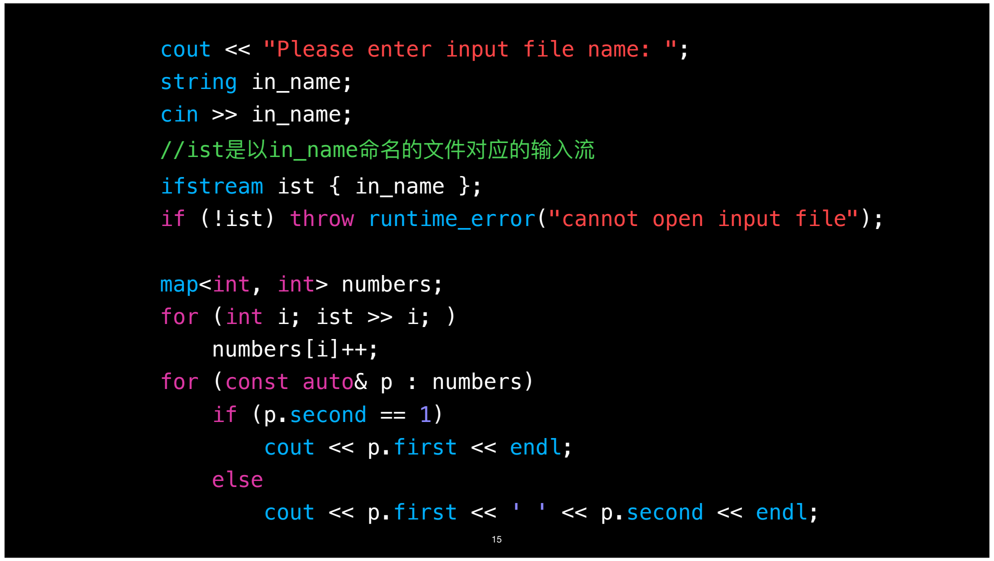
当ist和ost对象离开其作用域时会自动关闭所关联的文件



统计数值在文件中出现次数

- 编写程序，读取一个以空白符间隔的数值文件，按升序输出这些数值，每行一个值，每个数值只输出一次，如果一个数值在文件中出现多次，同时输出它出现的次数。例如，若原文件内容为“7 5 5 7 3 117 5”，那么输出

```
3
5 3
7 2
117
```



```
cout << "Please enter input file name: ";
string in_name;
cin >> in_name;
//ist是以in_name命名的文件对应的输入流
ifstream ist { in_name };
if (!ist) throw runtime_error("cannot open input file");

map<int, int> numbers;
for (int i; ist >> i; )
    numbers[i]++;
for (const auto& p : numbers)
    if (p.second == 1)
        cout << p.first << endl;
    else
        cout << p.first << ' ' << p.second << endl;
```